BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,
Please do not report the images to the
Image Problem Mailbox.

Group I.[2] Because the two Groups are "related" and properly searched in the same class and subclass, division is inappropriate.

Second, the two groups recite overlapping subject matter. A search of either of the two groups will all but inevitably result in a search of the subject matter of the other. There is no "serious search burden" in examining all these claims together.

Third, this disclosure has already been voluntarily divided into twenty-five applications. Because this fine division has already occurred, the claims of this application are directed to patentably-distinct but closely-related aspects of the same invention, and should be examined together.

Fourth, the Patent Office has now exceeded the statutory schedule for completing examination, and has now exceeded by a factor of three the statutorily-recommended schedule for first examination on the merits. The Patent Office should not further delay first examination on the merits of any non-elected claims. Restriction at this late date is inappropriate.

II. Overview of the Invention

Applicant offers this overview of the specification and figures in order to assist in selecting a more appropriate search class.[3]

Attention is drawn to Section III of the specification, pages 32-51 and **Figs. 3a-3n**. "Tapestry" is the name of the system described in the specification. A Tapestry processor has hardware for executing RISC instructions, and several different modes for creating and operating virtual Intel X86 (also known as "IA-32") machines to execute X86 programs. The ability to run programs in either of two instruction sets opens the possibility that a single program might be coded in both instruction sets, and use resources from both architectures.

The Intel X86 has only about 23 registers (depending on how one counts), most of which are only 32 bits wide. An operating system **306** for an X86 computer (such as Microsoft Windows or IBM OS/2) only has software capabilities for managing these 23 registers on a

---

[2] It will be understood that this introductory discussion of the invention and the claims (and analogous introductions to the other claims made in this paper) is an indication of the general search field most likely to be relevant to the claim, not a statement of the scope of the claims.

[3] One specific embodiment of the invention is discussed here, in order to provide context to assist the Examiner's search. This discussion is directed to assisting in framing an efficient search, and therefore is intended to direct the Examiner to the place where the best prior art is likely to be. It should be understood that this contextual discussion of one particular embodiment is not a limiting discussion of the invention or the scope of the claims.

context switch. In contrast, referring to **Fig. 3a** and to Table 1 at pages 21-22 of the specification, the Tapestry processor has 64 general registers, named "r0" through "r63," which are 64 bits wide. An unaltered, unassisted X86 operating system cannot context switch a 64-register process.

In the Tapestry System, X86 threads (*e.g.*, **302, 304**) carry the normal X86 context, including the X86 registers, as represented in the low-order halves of r32-r55, the EFLAGS bits that affect execution of X86 instructions, the current segment registers, etc. In addition, an X86 thread **302, 304** may embody a good deal of "extended context" (context state that is stored in the portion of the Tapestry processor context beyond the context of the X86 architecture), including the various Tapestry processor registers, general registers r1-r31 and r56-r63, and the high order halves of r32-r55 (see Table 1), the current value of ISA bit **194** (a bit that tells whether the hardware is currently executing in native Tapestry RISC mode or X86 mode), and other processor status and control information.

An unaltered, unassisted off-the-shelf X86 operating system **306** cannot manage threads **302, 304** that rely on extended context. Because X86 operating system **306** is coded in the X86 instruction set, and the X86 instruction set does not have instructions to access the extended context resources, the extended context cannot even be read or written by X86 operating system **306**, let alone switched on a context switch.

The Tapestry system performs some additional housekeeping on entry and exit to virtual X86 **310**, in order to save and restore the extended context, and to maintain the association between extended context information and threads **302, 304** managed by X86 operating system **306**.

Figs. 3a-3n describe one mechanism that can be used to save and restore the full context of an X86 thread **304** that is currently using Tapestry extended resources. In overview, this mechanism snapshots the full extended context into a memory location **355** that is architecturally invisible to virtual X86 **310**. A correspondence between the stored context memory location **355** and its X86 thread **304** is maintained by Tapestry virtualization system **312, 316** in a manner that that does not require cooperation of X86 operating system **306**, so that the extended context will be restored when X86 operating system **306** resumes X86 thread **304**, even if X86 operating system **306** performs several context switches among X86 threads **302** before the interrupted X86 thread **304** resumes. The Tapestry virtualization system **312, 316** briefly gains control at

each transition from X86 to Tapestry or back, including entries to and returns from X86 operating system **306**, to save the extended context and restore it at the appropriate time.

The details of that context saving and restoring are described in further detail in the specification, and in **Figs. 3i, 3j**, and **3k**. When a process is about to enter X86 operating system **306**, a save slot **355** (a block of storage sized to hold a full Tapestry context snapshot) is allocated (step **361**) from among those currently free. The entire Tapestry context, including the X86 context and the extended context, is saved (step **362**) into the context space **356** of allocated save slot **355**. The process context is then overwritten with information that enables the Tapestry virtualization system **312** to find the save slot – in other words, X86 operating system **306** never actually sees the context of the process that it is scheduling, only unintelligible "junk" cobbled up by Tapestry virtualization system **306**. X86 operating system **306** is then invoked to handle the interrupt.

Referring to **Fig. 3h**, section **370** shows resumption of a process after the X86 operating system **306** completes. The contents of the current process is examined to determine which process has been resumed by the X86 operating system. Of course, that process is not ready to execute – the context contains the Tapestry-specific "junk." Instead, Tapestry virtualization system **312**, **316** restores the correct context from the save slot, and then the process is allowed to resume.

III. Group II is Not Properly Divided From Group I Because The MPEP Instructs That "No Reasons Exist For Division" Between These Related Groups

Applicant traverses the Requirement for Restriction between Groups I and II. MPEP § 803 states the requirements for a restriction requirement (emphasis added):

> There are <u>two</u> criteria for a proper requirement for restriction between patentably distinct inventions:
> (1) The inventions must be independent (see MPEP §802.01, §806.04, §808.01) or distinct as claimed (see MPEP §806.05 - § 806.05(i)) <u>and</u>
> (2) There must be a <u>serious</u> burden on the examiner if restriction is not required (see MPEP §803.02, §806.04(a)-(j), §808.01(a) and §808.02).

MPEP § 803 clarifies (emphasis added):

> If the search and examination of an entire application can be made without <u>serious burden</u>, the examiner must examine it on the merits, even though it includes claims to distinct or independent inventions.

A. The January Restriction Requirement is Incomplete

Applicant concedes that Groups I and II are distinct (criterion (1) of MPEP § 803), but traverses under criterion (2). The Restriction Requirement omits any mention of criterion (2) — the Restriction Requirement makes a thorough showing of "independent and distinct," but is entirely silent on "serious search burden." At a minimum, a Restriction Requirement without such a showing is incomplete, and can neither be maintained nor made final.

B. Group II (claim 79-81): the Classification Proposed in the Restriction Requirement is Incorrect

The Restriction Requirement of January 2003 proposes to classify Group II in class 709, subclass 331. The Patent Office has already acknowledged that this classification is incorrect. Claim 79 recites as follows:

> 79. A method, comprising:
>
> during invocation of a service routine of a computer, passing a linkage return address to the service routine at which to resume execution on completion of the service, the linkage return address being deliberately chosen so that an attempt to execute an instruction from the linkage return address on return from the service routine will raise a program execution exception;
>
> on return from the service routine, attempting to execute the instruction at the linkage return address and raising the chosen exception; and
>
> after servicing the exception, returning control to a caller of the service routine.

In pertinent part, the definition for subclass 331 reads as follows:

> **310 INTERPROGRAM COMMUNICATION, INTERPROCESS COMMUNICATION (IPC):**
>
> This subclass is indented under the class definition. Subject matter comprising means or steps for exchanging data or messages between two executing programs or processes, independent of the hardware used in the communication.
>
> (2) Note. The subject matter of this subclass is directed to communication between processes.
>
> (6) Note. This subclass is for communication between processes and tasks, communication between computers or digital data processing systems and peripherals is classified elsewhere.
>
> **331 DYNAMIC LINKING, LATE BINDING:**
>
> This subclass is indented under subclass 310. ...

The previous Restriction Requirement, of August 2002, attempted to classify claim 79 in subclass 318. Both subclasses 318 and 331 are indented under subclass 310. Applicant's October 2002 paper showed that no subclass of subclass 310 could be appropriate.[1] This January 2003 Restriction Requirement acknowledges the correctness of Applicant's October showing. For exactly the same reason that claim 79 is not usefully searched in subclass 318, claim 79 is not usefully searched in subclass 331.

The embodiment of Group II described in the specification[2] relates to managing processes in a virtual machine emulation system (see pages 49-50 of the specification), the subject matter of class 709, subclass 1. Indeed, the title of the application nearly matches the class definition. The pertinent portion of the definition for subclass 1 reads as follows:

> **1 VIRTUAL MACHINE TASK OR PROCESS MANAGEMENT:**
>
> This subclass is indented under the class definition. Subject matter comprising means or steps operating on a computer or digital data processing system which enable a first type of processor to emulate and execute instructions associated with one or more different types of processors.
>
> (1) Note. This subclass is directed to subject matter encompassing one or more virtual machines that execute in single task, or multitasking, operating system environments.
>
> (2) Note This subclass includes computers or digital data processing systems executing a plurality of virtual machines that are preemptively or nonpreemptively scheduled.
>
> (3) Note. This subclass includes means or steps for mimicking the performance of one processing device within another processing device. For example, a software program that allows applications written for a first computer to be executed on a different second computer interpreting the machine instructions for the first computer, thereby becoming a virtual machine.

---

[1] To fall within subclass 310, a claim must recite communication between "two executing programs or processes." However, claim 79 is practiced when one single process invokes a service routine. There is no recitation of "two processes." Claim 79 cannot fall within subclass 310; therefore, claim 79 necessarily cannot fall within subclass 331.

Claim 79 could, conceivably, also be practiced by two processes, to the degree no limitation excludes the possibility. However, theoretical possibilities are not proper bases on which to classify claims for primary search.

[2] This is only the embodiment in the specification, not the scope of the claims. The claims may well be practiced in contexts outside class 709, subclass 1, or by techniques equivalent to those recited in the claims, and applicant has no objection to searching other potential subclasses, but that does not change the fact that the most efficient and effective search will be the search that focuses on the embodiment that is the subject of the specification and many of the dependent claims. The primary search – and the classification for restriction purposes – should be the subclass most related to the embodiments in the disclosure, and claimed in the dependent claims.

Claim 81, one of the dependent claims in Group II, clarifies that subclass 1 is the most appropriate search subclass for Group II as a whole:

> 81. The method of claim 79, wherein
> the service routine is an interrupt service routine of an operating system for a computer architecture other than the architecture native to the computer;
> the service routine is invoked by an asynchronous interrupt; and
> the caller is coded in the instruction set native to the architecture.

Handling an operating system that is not native to the computer is properly searched as "Virtual Machine Task Or Process Management," the subject matter of subclass 1.

The title, the specification (*see, e.g.*, specification, page 26, line 8; page 27, line 10; page 33, lines 3, 8, 9, and 18; page 40, lines 10, 20 and 23, all showing that the specification is directed to implementation of a "virtual machine" or "virtual X86"), and dependent claims 80 and 81 clarify that prior art relevant to Group II, if it exists at all, is much more likely to be found in subclass 1 than subclass 331. A search of subclass 331 has a significant risk of being entirely ineffective with respect to any claim in Group II.

For these reasons, Applicant suggests that Group II should be searched in class 709, subclass 1 instead of subclass 331. Subclasses 102, 318 and 331 may well turn out to be appropriate subclasses into which to expand a secondary search, but none of these can be the best primary search class.

C. As Correctly Classified for Search, Groups I and II are Properly Examined Together

The Restriction Requirement concedes that Groups I and II are "related" (paragraph 3). "Related" groups may not be restricted when they are classified together for search. MPEP § 808.02 ("Where, however, the classification is the same and the field of search is the same and there is no clear indication of separate future classification and field of search, no reasons exist for dividing among related inventions."). Because all claims should be searched in the same search class and subclass, no reasons exist for restriction.

IV. A Search of Group I Will Inevitably Involve A Search of Group II

Any search burden is unlikely to be "serious" when claim 79 (the independent claim of Group II) appears nearly word-for-word in claim 83 of Group I. Claims 5 and 20 are part of

Group I and must be searched in any event, and claim 83 must be searched unless claims 5 and 20 are given a first action allowance. Therefore, a search of Group I will all but necessarily result in a search of the independent claim of Group II as well.

Similarly, a search of Group I will involve a search of claim 52. Claim 52 recites subject matter that, at least for search purposes, closely overlaps the subject matter of claims 79-81 (the claims of Group II).

Thus, a search of Group I (the group elected) will all but inevitably involve a search of Group II. Because no "serious" burden exists in searching Group II simultaneously with Group I, the claims are appropriately examined together.

V. Further Facts Suggest That Restriction is Inappropriate

This disclosure has been voluntarily divided into 25 applications. The 24 other applications are listed in the Information Disclosure Statement filed January 17, 2001. Because of this fine division, it should be apparent that the claims in this application are closely related, and that further division is unwarranted.

Further, this application has now been pending for over four years and has yet to receive its first action on the merits. 35 U.S.C. § 154(b)(1)(A)(i) indicates Congressional intent that an application receive its first consideration within fourteen months of filing, and that prosecution conclude within three years. If this restriction requirement is affirmed, the divided claims may well not receive their first consideration for years more. Even now, these delays are far in excess of the statutorily-prescribed guideline for PTO action. Failure to act within this time is "unreasonable delay," and is in conflict with the PTO's legal obligations under the Administrative Procedure Act. 5 U.S.C. § 555(b); *Telecommunications Research and Action Center v. Federal Communications Commission*, 750 F.2d 70, 80 (D.C. Cir. 1984) ("The time agencies take to make decisions must be governed by a 'rule of reason,' [and] where Congress has provided a timetable or other indication of the speed with which it expects the agency to proceed ..., that statutory scheme may supply content for this rule of reason.").

If division of this application was ever proper, the time for doing so is long past. In view of the undue delay on the part of the Office, far in excess of statutory guideline, and substantial loss of patent term occasioned thereby, it would be entirely inappropriate for the PTO to now divide the application on anything less than the clearest showing of proper division of the

application. If grounds for division validly exist at all, they are tenuous at best. No division should be required.

Applicant requests that the application be passed to issue in due course. The Examiner is urged to telephone Applicant's undersigned counsel at the number noted below if it will advance the prosecution of this application, or with any suggestion to resolve any condition that would impede allowance. Enclosed is a Petition for Extension of Time for one month. In the event that further extension of time is required, Applicant petitions for that extension of time required to make this response timely. Kindly charge any additional fee, or credit any surplus, to Deposit Account 50-0675, Order No. 5231.5-4013.

Respectfully submitted,
SCHULTE ROTH & ZABEL, LLP

Dated: February 6, 2003

By: _____
David E. Boundy
Registration No. 36,461

Mailing Address:
SCHULTE ROTH & ZABEL, LLP
919 Third Avenue
New York, New York 10022
(212) 756-2000
(212) 593-5955 Telecopier